

ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ ОПЕРАТОРОВ ЯЗЫКА МОДЕЛИ ДАННЫХ

В.И. ЕСИН, М.В. ЕСИНА

Описывается проблема использования различных языковых средств по поддержанию ряда функций интероперабельных информационных систем, которая может быть решена с помощью специального языка модели данных. Приводятся возможные варианты использования операторов этого языка и даются некоторые рекомендации по их применению.

Ключевые слова: модель данных, универсальная модель данных, язык модели данных, база данных.

ВВЕДЕНИЕ

Анализ использования различных языковых средств [1-5] по поддержанию ряда функций интероперабельных информационных систем, связанных с представлением знаний о предметной области (необходимых для разработки формализованных описаний при концептуальном моделировании); созданием программных приложений; разработкой средств семантического согласования; разработкой средств разграничения прав доступа к данным и т. д., показал что они:

- как правило, ориентированы на поддержку только отдельных выше названных функций;
- требуют знания модели (схемы) базы данных или структуры источника данных;
- не в состоянии стать подходящим средством выражения семантики данных и способствовать решению задачи автоматической трансформации семантически правильного запроса, составленного в терминах предметной области (ПрО) с использованием естественного языка, в синтаксически и терминологически корректный запрос к конкретной базе данных;
- достаточно сложны в освоении не специалистами в области информационных технологий.

Все это послужило поводом для разработки специального языка модели данных, лишённого указанных выше недостатков. И таким языком стал язык модели данных (ЯМД) [6].

В работе [6] уже говорилось о ЯМД как о специальном непроцедурном (декларативном) основывающемся на метаонтологиях модели «объект-событие» языке, который позволяет определять метаданные и данные любой предметной области, а также манипулировать ими в терминологии близкой к естественному языку. Но операторы (точнее строки метаописания, составленные из операторов) ЯМД могут использоваться не только для описания элементов рассматриваемой ПрО и манипулирования данными, занесёнными в базу данных (БД) с универсальной моделью данных (УМД). Они могут применяться и в других различных целях, способствующих эффективному и корректному использованию данных, хранящихся в различных источниках. Рассмотрим эти возможные варианты.

ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОВ ЯМД

1. Операторы ЯМД используются в соответствующих информационных системах (ИС), когда необходимо реализовывать механизм распределения прав доступа к данным вплоть до любого конкретного элемента данных.

С этой целью в рамках ядра схемы БД с УМД была создана таблица «Ограничения прав доступа» к конкретному элементу данных. На псевдокоде ее можно описать следующим образом:

```

USER_ID INTEGER not null, -- Идентификатор
                             пользователя
USER_NAME VARCHAR(255) not null -- Имя
                             пользователя
CLASS_ID INTEGER not null, -- ID-объекта
                             в таблице TABLE_NAME
TABLE_NAME VARCHAR(255) not null, -- имя
                             таблицы, для которой создается
                             ограничение прав доступа
DESCRIPTOR CLOB not null -- строка
                             метаописания ЯМД
primary key (CLASS_ID, TABLE_NAME) --
                             РК-первичный ключ.

```

В данной таблице в соответствующем атрибуте (DESCRIPTOR) хранится строка метаописания языка модели данных. На основании значения этого атрибута определяется адрес защищаемого элемента (его идентификатор *ID* в требуемой таблице БД с УМД), доступ к которому ограничивается. Процесс заполнения атрибутов данной таблицы автоматизирован с помощью специального программного приложения, использующего в качестве основного входного параметра именно ту строку метаописания, которая задает адрес защищаемого элемента данных.

Поясним данный механизм на примере ограничения прав доступа пользователя «AVT1» к экземпляру объекта «333333» класса «Техническое средство» и типа «Вольво», принадлежащего пользователю «AVT2».

Вначале составляется строка метаописания, по которой однозначно можно определить адрес данного экземпляра объекта в БД с УМД. В нашем случае вид этой строки будет следующий:

```
{<Раздел>=Хар_УГЭС; /<КлассО>=Техническое
средство; <ТипО>=Вольво; <ЭкзО>=333333;}
```

Затем запускается на выполнение специальная программа определения прав доступа к данным. При этом подключение в ней к БД с УМД

должно осуществляться с правами того пользователя, данные которого он хочет ограничить в использовании другому пользователю (рис. 1).

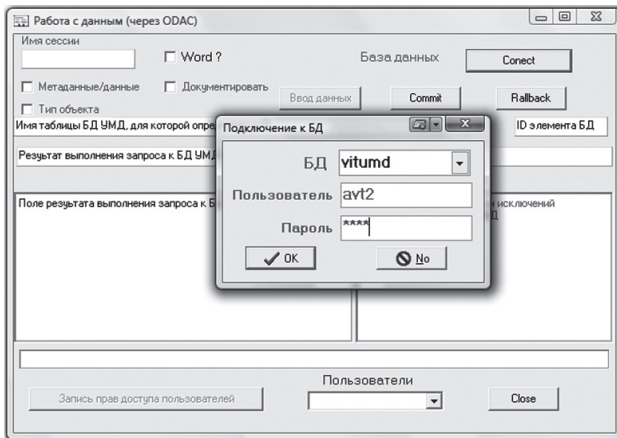


Рис. 1. Программа задания прав доступа к данным

В нашем случае пользователь «AVT2» ограничивает использование экземпляра объекта «333333» для пользователя «AVT1». Для этого он (пользователь «AVT2») с помощью вышеуказанной программы выбирает в окне «Пользователи» пользователя «AVT1» и исполняет приведенную выше строку метаописания ЯМД с фиксацией «кнопки» «Запись прав доступа пользователей» (рис. 2).

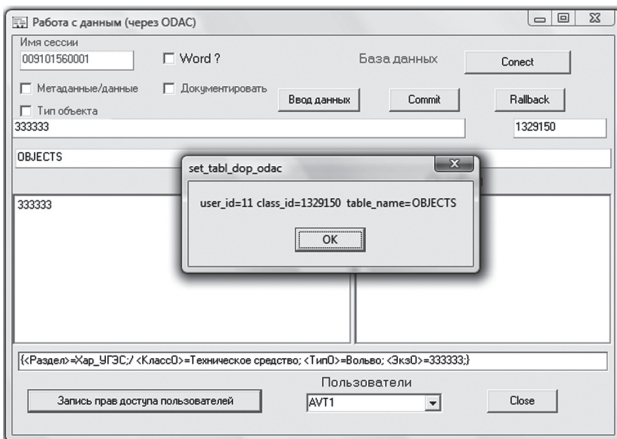


Рис. 2. Ограничение прав доступа пользователя AVT1

После чего данные полей приложения: «AVT1», «OBJECT», «1329150», «{<Раздел>=Хар_УГЭС;/ <КлассО>=Техническое средство; <ТипО>=Вольво; <ЭкзО>=333333}» автоматически записываются в таблицу ограничения прав доступа, в соответствующий ее атрибут. Просмотр содержимого таблицы с помощью запроса

```
SELECT * FROM "Ограничения прав доступа"
WHERE CLASS_ID=1329150
```

покажет следующий результат:

USER_ID	CLASS_ID	TABLE_NAME	DESCRIPTOR	USER_NAME
11	1329150	OBJECTS	{<Раздел>=Хар_УГЭС;/ <КлассО>=Техническое средство; <ТипО>=Вольво; <ЭкзО>=333333;}	AVT1

На основе данных этой таблицы, реализованные в схеме БД с УМД политики (речь о которых шла в [7]) автоматически обеспечивают защиту соответствующих данных от несанкционированного использования. Теперь если пользователь «AVT1» захочет обратиться из любого приложения (будь то Oracle SQL*Plus, PL/SQL Developer либо любое другое) к данным БД с УМД, то экземпляр объекта «333333» ему будет не доступен, так как доступ к нему «закрыт» на уровне сервера СУБД.

В подтверждение сказанному ниже приведены обращения пользователей «AVT1» и «AVT2» к БД с УМД, выполненные с помощью различных приложений. Так на рис. 3, 4 приведены данные, выводимые специальным приложением программного инструментария разработчика БД с УМД – программой просмотра, для пользователей «AVT1» и «AVT2» соответственно.

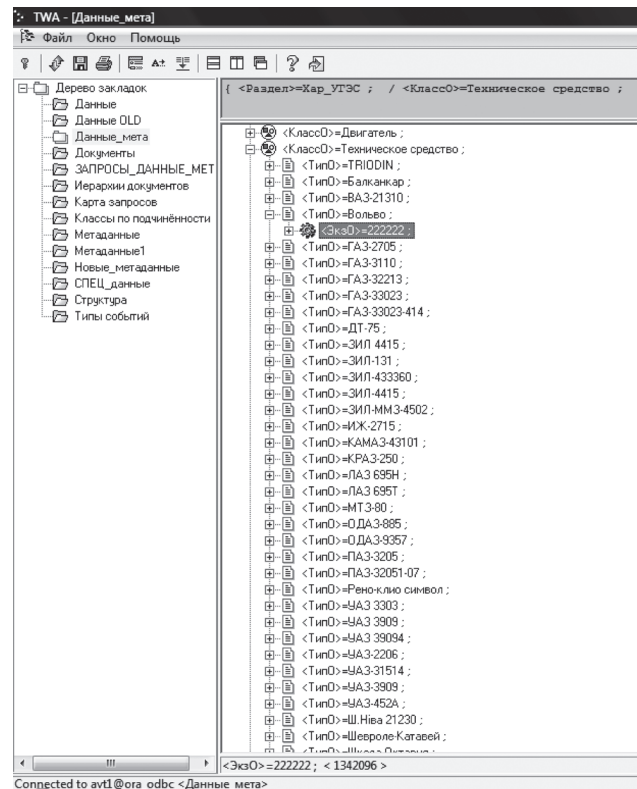


Рис. 3. Данные, выводимые программой просмотра для пользователя AVT1

Как видно из этих рисунков, пользователю «AVT1», в отличие от пользователя «AVT2», не доступен экземпляр объекта «333333».

При обращении к данным в среде PL/SQL Developer с помощью следующего кода:

```
declare
id number;
pr_type number;
text_vx_str varchar2(5000);
tmp_char varchar2(2000);
cur_name_end_text data_umd_big.my_ref_cursor;
i_count number;
i number;
name_session_proc varchar2(255);
```

```

text_exit varchar2(250);
error_text varchar2(5000);
pr_write number;
begin
pr_type:=1; id:=0; i_count:=0; pr_write:=0;
text_vx_str:=' {<Раздел>=Хар_
УГЭС;</КлассО>=Техническое средство;
<ТипО>=Вольво; <ЭкзО>=*. *?;}';
DATA_UMD_BIG_P(text_vx_str, pr_type, pr_
write, id, name_session_proc, text_exit, er_
ror_text, cur_name_end_text);
i:=0;
LOOP
fetch cur_name_end_text into tmp_char;
exit when cur_name_end_text%NOTFOUND;
i:=i+1;
dbms_output.put_line('name'||('||i||')='||tmp_
char);
END LOOP;
end;

```

результат для пользователя «AVT2» будет:
name (1)=222222
name (2)=333333,
а для пользователя «AVT1»:
name (1)=222222.

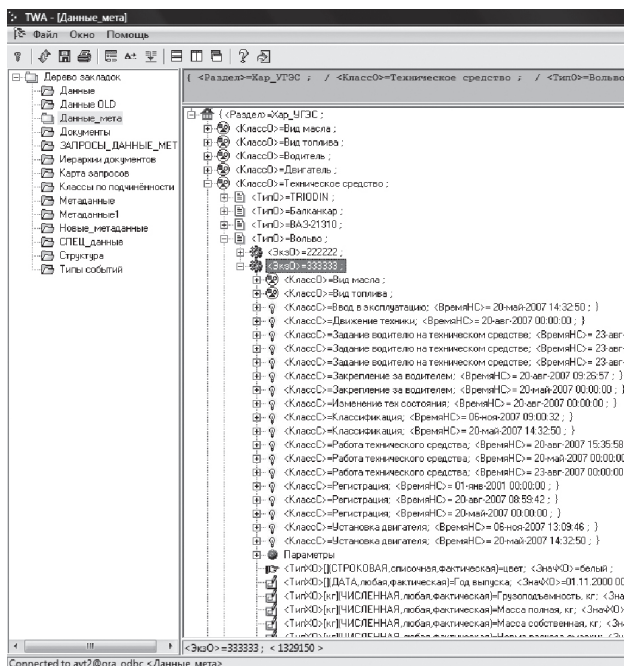


Рис. 4. Данные, выводимые программой просмотра для пользователя AVT2

Результат обращения пользователя «AVT1» к БД с УМД в среде Oracle SQL*Plus приведен на рис. 5.

Как видим результат один и тот же. Пользователь «AVT1» не имеет доступа к экземпляру объекта «333333». Если даже обратиться напрямую к таблице экземпляров объектов посредством оператора SELECT языка запросов SQL, результат для пользователя «AVT1» будет таким же – объект «333333» ему недоступен.

Аналогичным образом можно ограничить доступ различным пользователям к любым метаданным и данным любой предметной области.

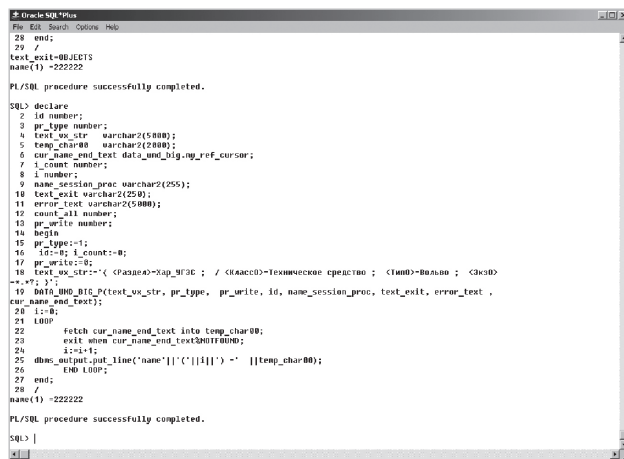


Рис. 5. Результат исполнения кода для пользователя AVT1 в среде Oracle SQL*Plus

2. Сегодня можно говорить, что эра, когда разработчики информационных систем приходили в организацию и начинали проекты информатизации «с нуля», прошла. Большинство организаций уже имеет некоторые информационные системы, которые со временем становятся бременем компании и начинают требовать реинжиниринга. Это объясняется тем, что специалистам любой крупной и давно существующей компании, которая обладает несколькими базами данных, относящимися к разным видам деятельности, данные в которых могут иметь разные представления и быть даже несогласованными, становится очень трудно связывать и анализировать содержащуюся в них информацию. К тому же многие базы данных, составляющие основу таких ИС, как правило, построены на уже «устаревших платформах» (например, Dbase, FoxPro и т. д.). При этом данные, в них хранящиеся, имеют большую практическую ценность.

Как результат, имея самые разнообразные изолированные друг от друга источники данных, найти необходимые для деятельности компании данные становится невозможным. Поэтому чтобы упростить и ускорить доступ к такой информации, дать возможность проследить связи между разными источниками данных и обеспечить надежность использования унаследованных настольных баз данных в обеспечении управления на всех уровнях, необходимо каким-то образом интегрировать эту информацию.

Сегодня организации обычно расходуют от 20 до 40% своего IT-бюджета на эволюционирование своих данных путем миграции (изменение местоположения данных), преобразования (изменения формы или структуры данных) или очистки (изменение или повторный ввод данных для последующего использования) [8].

А раз подобные задачи встраивания устаревших информационных компонентов в систему, основанную на новой технологии, сегодня достаточно часто приходится решать, то нужно сделать так, чтобы они были разрешимыми. То есть, чтобы компоненты унаследованных систем

сохраняли интероперабельность. Потому что, чем дольше живет и приносит пользу информационная система, тем это выгоднее для компании. Естественно, что для этого должна существовать возможность добавления в нее компонентов, спроектированных и разработанных, вообще говоря, в другой технологии. Сегодня в мире существует большое количество подходов, методов и технологических решений, напрямую или косвенно соотносимых с деятельностью по реинжинирингу ИС. Однако они не интегрированы на уровне методологий (процессов разработки). Как результат, можно наблюдать наличие огромного их количества, где основной акцент сделан на опять же разработку ИС «с нуля», и практическое отсутствие «строительных» методологий, целью создания которых являлось бы комплексное, целостное решение задач реинжиниринга ИС [9].

С этой целью в свое время был предложен метод решения этой проблемы [10], который в основе своей опирается как раз на использование ЯМД. Строки метаописания ЯМД используются в нем при конвертации данных из БД других информационных систем, построенных на различных платформах (таких как Oracle, PostgreSQL, Access), в БД с УМД. Более подробно о данном механизме излагалось в [10].

3. Строки метаописания ЯМД также могут быть использованы при создании отчетных документов, формирующихся, с помощью различных средств, в том числе и OLAP-технологии.

На рис. 6 приведен интерфейс приложения, использующего строки метаописаний ЯМД, для создания отчетных документов в форматах: PDF, RTF, HTML, XLS, CSV, XML, Jasper Reports, JRXML.

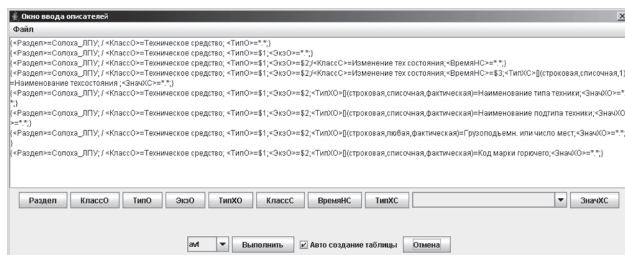


Рис. 6. Интерфейс приложения для формирования отчетных документов

Формы некоторых отчетных документов, полученные с использованием возможностей данного приложения, приведены на рис. 7, 8.

4. Сегодня для балансировки нагрузки на телекоммуникационные и вычислительные ресурсы распределенной системы очень часто используют механизм репликаций. И поэтому, чтобы обеспечить наиболее простую организацию этого механизма, при использовании баз данных с УМД вместо моментальных снимков основных таблиц (*snapshot*) БД асинхронно распространяется (реплицируется) таблица-журнал измененных данных (табл. 1). Из которой впоследствии

извлекаются и выполняются строки метаописаний ЯМД, что позволяет существенно снизить нагрузку на систему и ее трафик.

Рис. 7. Форма отчетного документа в Jasper Viewer



Рис. 8. Диаграмма распределения технических средств по типам

Таблица 1

Пример таблицы-журнала измененных данных

Системное имя	IP-адрес компьютера	Имя пользователя	Имя БД	Строка метаописания ЯМД	Данные или метаданные	Операция*	Время записи в текущую БД	Время выдачи данных в БД распределен. системы
VIT\vitaly	195.110.100.1	SHEMA_1	Org_umd	{<Раздел>=Нов_1;}	метаданные	insert	16.09.2008 10:00:01	17.09.2008 20:10:11
Nik\mucmp	101.63.102.28	SHEMA_2	spumd	{<Раздел>=Хар_УГЭС;/<КлассО>=Тех. средство; <ТипО>=Вольво; <ЭкзО>=111; <УдалитьО>=;}	данные	delete	26.07.2010 17:41:48	29.07.2010 17:41:48

* – операция чтения (*select*) не отслеживается.

При этом автоматически решается проблема целостности транзакций: внесение изменений в

реплицируемую БД с УМД осуществляется в ней же самой, путем исполнения переданных строк метаописаний ЯМД.

ВЫВОДЫ

1. Рассмотренные различные варианты использования строк метаописания ЯМД способствуют эффективному и корректному использованию (в соответствии с распределением прав доступа вплоть до любого конкретного элемента) данных, хранящихся в различных источниках.

2. Для того чтобы предложенные варианты использования строк метаописания ЯМД стали на самом деле эффективным средством в руках различных пользователей, последние должны иметь (сформировать) правильное представление о ПрО, которое затем следует адекватно отобразить в схеме БД с УМД. Независимо от того было получено это отображение в процессе первоначального моделирования базы данных или при переносе ее из других источников информации в схему БД с УМД.

Литература

- [1] Калиниченко Л. А. СИНТЕЗ: язык определения, проектирования и программирования интероперабельных сред неоднородных информационных ресурсов / Л. А. Калиниченко. — М. : ИПИ РАН, 1993. — 121 с.
- [2] Шрайбман В. Выражение семантики данных. RDF против XML [Электронный ресурс] / В. Шрайбман. — Режим доступа : http://citforum.ru/internet/xml/rdf_xml/
- [3] RDF схема метаданных ИСИР / [Бездушный А. А., Бездушный А. Н., Жижченко А. Б. и др.] // Современные технологии в информационном обеспечении науки — М. : Научный Мир, 2003. — С.141–159.
- [4] Токмаков Д. И. Использование средств языка RDF в аннотировании Интернет-ресурсов / Д. И. Токмаков // Информационные ресурсы России. — М. : Федеральное государственное учреждение Российское энергетическое агентство Министерства энергетики Российской Федерации, 2007. — № 5. — С.30–33.
- [5] RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation. [Electronic resource] — Access mode : <http://www.w3.org/TR/rdf-schema/>
- [6] Есин В. И. Язык для универсальной модели данных / В. И. Есин, М. В. Есина // Системы обработки информации. — Х. : Харьковский университет Повітряних Сил, 2011. — № 5(95) — С.193–197.
- [7] Есин В. И. Особенности защиты данных в базах данных с универсальной моделью / В. И. Есин, М. В. Есина // Прикладная радиоэлектроника. — Х. : Харьковский национальный университет радиоэлектроники, 2011. — Т. 10, № 2 — С.226–232.
- [8] Кузнецов С. Модель зрелости для управления данными [Электронный ресурс] / С. Кузнецов — Режим доступа : <http://www.citforum.ru/computer/2007-04/>
- [9] Ахтырченко К. В. Методы и технологии реинжиниринга ИС / К. В. Ахтырченко, Т. П. Сорокваша // Труды Института системного программирования РАН. / Под редакцией чл.-корр. РАН В. П. Иванникова. — М. : ИСП РАН, 2003. — Т. 4. — С.141–162.
- [10] Сорока Л.С. Реинжиниринг существующих баз данных в базу с универсальной моделью данных и возможностью распараллеливания процессов / Л. С. Сорока, В. И. Есин, М. В. Есина // Сборник научных трудов Параллельная компьютерная алгебра: Всероссийская научная конференция с элементами научной школы для молодежи, г. Ставрополь, 11-15 октября 2010 г. Ставропольский государственный университет. — Ставрополь: Издательско-информационный центр «Фабула», 2010, С.298-304.

Поступила в редколлегию 29.03.2012

Есин Виталий Иванович, кандидат технических наук, доцент, доцент кафедры безопасности информационных систем и технологий ХНУ им. В.Н. Каразина. Область научных интересов: модели и методы разработки баз данных информационных систем и обеспечение их безопасности.



Есина Марина Витальевна, студентка факультета компьютерных наук ХНУ им. В.Н. Каразина. Область научных интересов: базы данных и обеспечение их безопасности.

УДК 004.652: 004.655: 004.432.4

Варіанти використання операторів мови моделі даних / В.І. Єсін, М.В. Єсина // Прикладна радіоелектроніка: наук.-техн. журнал. — 2012. — Том 11. № 2. — С. 288–292.

Описується проблема використання різних засобів по підтримці ряду функцій інтероперабельних інформаційних систем, яка може бути вирішена за допомогою спеціальної мови моделі даних. Наводяться можливі варіанти використання операторів цієї мови і даються деякі рекомендації по їх вживанню.

Ключові слова: модель даних, універсальна модель даних, мова моделі даних, база даних.

Табл. 01. Іл.08. Бібліогр.: 10 найм.

UDC 004.652: 004.655: 004.432.4

Variants of using data model language operators / V.I. Yesin, M.V. Yesina // Applied Radio Electronics: Sci. Journ. — 2012. Vol. 11. № 2. — P. 288–292.

The problem of using different language means of supporting some functions of interoperable informative systems is described. This problem can be solved by means of a special data model language. Possible variants of using operators of this language and some recommendations for their use are given.

Keywords: data model, universal data model, data model language, database.

Tab. 01. Fig. 08. Ref.: 10 items.